

Ambisuite/Ambman: a utility for transforming ambisonic files

Michael CHAPMAN

place Cotter
01350 Culoz,
France,
chapman@mchapman.com

Abstract

The creation of a utility for converting and transforming ambisonic files is described.

The challenges of making such a utility as independent as possible of the user's platform and of adding a GUI to what was initially a command line tool are detailed.

Though the original version was released over a year ago, a greatly revised version of Ambisuite (*ver. 0.6.0*) is due to be released to coincide with LAC2009.

Keywords

Ambisonics, Ambisuite, Ambman, batch processing.

1 Introduction

Ambisonic¹ files are multi-channel audio files. They may, though, contain any number of channels (from a trivial one or two, but more realistically from three channels to an infinite number).

The user of such files is confronted with two main problems:

- Converting from one file format to another
- Transforming [1] a signal set –for example rotating the soundfield

The utility described uses a PERL [2] script as a wrapper to access the functionality of SOX [3] to allow such conversions and manipulations.

1.1 Ambisonic file formats

1.1.1 Classic formats

The commonest format for ambisonics files, at the time of writing, is a form of WAVEX (Wave Format Extensible), which has a specific GUID² (the equivalent of a UUID) and the suffix `.amb`.

The format suffers from two main limitations:

¹See *Appendix* (p. 6) for a brief description in relation to the terms used here.

²Acronyms are generally defined on first occurrence in the body of the text. Common ones, which are not, are listed/defined in section 6.

– The limit on `.wav` file size³

– The non-explicit identification of channels

For first order ambisonics the first is generally acceptable (a four channel ambisonic file has half the play time of a stereo file at the same sampling rate and bit-depth). The latter is circumvented by restricting usage to the first three orders (where channel count can be correlated to the format of the signal set⁴).

Its great attraction has been the ubiquity of applications that handle `.wav`. Despite that, applications that recognise `.amb` files are few, if any, and users have complained of having to rename the file (changing the suffix to `.wav` to use files).

1.1.2 The need for a format to include higher order material

There is an obvious need for a format that does not suffer (practical) file size limits and in which channels can be explicitly identified.

From Ambisuite 0.6.0, a `.caf` format [5] is offered with its own specific UUID:

“5dc3f270c2d24293858e64da38090bea”

which also has a metadata chunk. This is more as ‘proof of concept’ than as a statement that *this* should be the new format. Comments from users are invited.

The same UUID can be used in `.wav` files (implying an obligatory metadata chunk) allowing the same functionality for non-large files. This allows the lossless compression of WavPack [6] to be exploited.

³4GB (though 2GB in some implementations), that is about 2 hours of 48KHz/24-bit 4 channel audio. Variants allow for a 64-bit file size description (as against 32) that remove this problem for practical purposes (‘W64’ and ‘RF64’) as does using metadata to chain files.

⁴See [4]. If more complex mixed-order signal sets come into use (e.g. (H, V, P) sets, see page 6) then the situation is more complicated, and these could not all be implicitly determined.

1.1.3 The need for meta-data

The trial `.caf` format (and related `.wav` files) use a chunk in XML to store metadata describing the file content. The DTD at:

`ambisonics.ch/dtds/ambisonic-0.03.dtd` has been refined after discussion on the Ambisonics Association mailing list.

The DTD countenances files in publication/exchange format ('B-format') as well as as unprocessed microphone signals, the UHJ broadcast formats, and ambisonics decoded to loudspeaker feeds (so called 'A-', 'C-' and 'D-format's). For B-format, of primary relevance to Ambisuite, it allows metadata to be stored on whether the signal set represents a two-, three-, mixed-, etc. dimensional soundfield and the assignment of the channels of the file to ambisonic channels. It is intended that its style will be able to include developments in ambisonics without breaking backwards compatibility.

It has been commented [7] that OSC [8] may prove to be a better alternative than XML. Though generally associated with dynamic control of audio, OSC has been used for static configuration files [9]. (There has also been talk – no more – of dynamically controlling SoX using OSC, which would be very interesting for ambisonic usage: for example to alter the aspect of (to rotate) a soundfield whilst it is playing.)

1.2 Ambisonic manipulations

The basic manipulations (other than adjusting amplitude/volume) are rotation and mirroring. A *yaw* (an anti-clockwise rotation about the *z*-axis) is easily achieved for a signal set of any order, likewise mirroring (*shake*). The other rotations (*pitch*, *roll*) have complicated transformation matrices and thus far are only implemented upto second order. Extension to higher orders will be accomplished along with generalised rotations (that is not restricted to the planes of the axes). This though is a project for the future, indeed it has only just become practicable with Franz Zotter's publication [10] of matrices upto twenty-first order, earlier this year. These are currently used in *shake* to allow simple 90° step rotations.

Dominance ([11] p.5: a form of zooming) is only possible (for theoretical reasons) for first order files ([12] pp. 7-8).

A mono sound can also be *placed* at a specific polar coordinate. The ability to do this is limited by the available spherical harmonic equations, that is upto sixteenth-order.

A series of dynamic manipulations are being developed, e.g. such that *yawing* gradually rotates a signal set from one position to another.

The other manipulations are of niche interest and documented in the software (e.g. `ambman --help=a2b`).

manipulation	order
a2b	1*
b2a	1*
mix	∞
normalise	∞
pitch	2
pitching	0
place	16
roll	2
rolling	0
shake	21
wxy2amb	1*
yaw	∞
yawing	4

Table 1: Manipulations offered by Ambman. The limits marked '*' are the maximum allowable by ambisonics, rather than software limits. ('0' is not yet implemented.)

2 Implementation

2.1 Minimal system calls

As the aim was for universality, then system calls have had to be curtailed, and the functionality of shell commands not used. Basic file and directory operations are carried out using Perl's 'filehandles, files and directories' functions. The only call to Perl's `system` function is to SoX and all such calls occur through one subroutine to ensure systematic error reporting should SoX fail (due to lack of discspace, a corrupt input file, or whatever).

2.2 Extensibility

The aim has been to create 'any-order' utilities. This meant abandoning classical channel sequence and channel weightings (the FuMa or Furse-Malham format [13]). That format is only applicable to the first three orders (upto sixteen channels), and its unsystematic approach makes software writing both difficult and prone to errors. Instead signals are in a normalised format (N3D ([14] pp. 155-157, esp. table 3.2 on p. 156)) and handled in the sequence of their ambisonic channel number (ACN) [15].

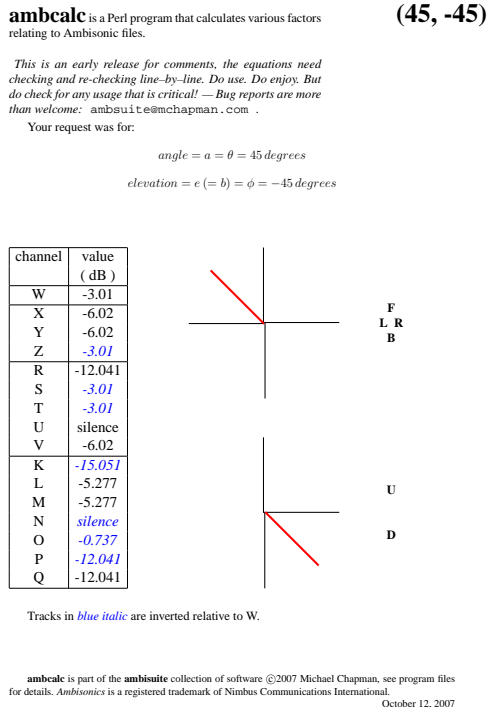


Figure 1: PDF output (in FuMa) of Ambcalc.

2.2.1 Keeping it modular

The aim has been that all data in the code (e.g. ‘look up tables’, equations) occur once only and then used if necessary by multiple subroutines. This eases both error detection and correction. All basic code is in a series of subroutines, whilst this makes the code verbose, it does facilitate later editing.

2.3 Adding a gui

At the time of writing a GUI is being developed using the Tk toolkit [16] (version 8.5) accessed with the Perl Tkx package [17]. Conceptually this is quite different from batch processing and has resulted in much code having to be rewritten. (The greatest work has been in having to change the promiscuous warning and error messages that were output to terminal to systematic ones that could be rendered on either terminal or GUI.)

It is an exciting challenge, and one that is probably essential if Ambisuite is to reach musicians, composers and others who may be averse to command line syntax.

3 Usage

Whilst not the prime purpose of this paper, it would be remiss not to give a brief guide to using the utilities.

The compressed bundle can simply be opened in a directory and used there. To get the maximum benefit they should be installed (with root privileges) and then can be used anywhere. All the *.pl files (except libambisuite.pl) should be moved to a directory on the users’ search paths (/bin/ works fine on most systems). Renaming them by dropping the suffix (e.g. `cp ambman.pl /bin/ambman`) makes for easier use, and the GUI script can be renamed (i.e. `cp ambisuite-gui.pl /bin/ambisuite`). Permissions should already allow for execution but can easily be changed if necessary (e.g. `chmod 755 /bin/ambman`).

The library of sub-routines (libambisuite.pl) needs to be in Perl’s search path (and should *not* be renamed). Trying to execute a mythical Perl program will cause Perl to list its search path (e.g. `perl -e 'require "zyxw.pl";'`), and one of the listed directories can then be used as a location for ambisuite.pl. (E.g. /usr/lib/perl5/vendor_perl/.)

The utilities can then be called in a terminal. Calling without options will give basic usage information, or they can be called with the option `--help`. (Sample terminal outputs for informational utilities are in figures 2, 3, 4 and 5.) As yet there is no icon for opening the GUI, but typing **ambisuite** and then RETURN in a terminal will bring up the introductory window.

```

]$ ambcalc
AMBCALC 0.5.3, 2007/09/18 - Copyright (c) 2007 by Michael

This is an early release for comments, the equations need
checking and re-checking. Do use. Do enjoy. But do check
for any usage that is critical! --- Bug reports are more
than welcome: ambsuite@mchapman.com

type q to quit, h for help

angle      (a): 45
elevation  (e): -45
channel    i value ( dB )
-----
W          -3.01
X          -6.02
Y          -6.02
Z          -3.01
-----
A "-" in the "i" column indicates the track is inverted
relative to the W track (that is the amplitude is
negative).
```

Figure 2: Terminal output (in FuMa) of Ambcalc.

```

]$ ambchan hv=4,2
AMBCHAN 0.5.3, 2008/09/18 - Copyright (c) 2008 by Mich

Channels for (H,V) = (4,2):
(Malham notation = ffhh.)

0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 15 , 16 , 24
Total channels used = 13.

1  m ACN
0  0  0
1 -1  1
1  0  2
1  1  3
2 -2  4
2 -1  5
2  0  6
2  1  7
2  2  8
3 -3  9
3  3 15
4 -4 16
4  4 24

```

Figure 3: Terminal output of Ambchan, giving channels in a $(h, v) = (4, 2)$ file.

```

]$ ambchan R
AMBCHAN 0.5.3, 2008/09/18 - Copyright (c) 2008 by

AMBISONIC CHANNEL:
6 ---ambisonic channel number (ACN)
2,0 --- (l,m) -- (order,range)
R ---channel letter (if appropriate).

The formula for the spherical harmonic associated
with this second degree component can be found at:
http://ambisonics.ch/standards/channels/ACN6

```

Figure 4: Terminal output of Ambchan, giving synonyms for channel R .

```

]$ ambinfo MC_26-positions.amb
AMBINFO 0.5.4, 2008/10/31 - Copyright (c) 2007-8 by Michael

SUMMARY; (run with -v for more details)
Channels: 4

Implicitly a first order file
channels (I am guessing) are WXYZ

48 KHz / 24 bits / PCM / AMBISONIC
estimated duration 130 seconds:

```

Figure 5: Terminal output of Ambinfo.

4 The Future

There is still much work to be done, even in getting the present features running smoothly. User feedback is welcomed to assist with that process.

Whilst a ‘modern’ file format has been proposed in the present release of Ambisuite, this is there very much as an example of how things

may be.⁵ If the ambisonic community can come together behind a file format, that will need incorporating. Most of the coding is though already there and adding a new format should now be relatively easy.

The question of mixed-order files (see page 6) is somewhat unstable at present. Ambisuite supports classical mixed order files, described by order (highest degree) and highest periphonic degree. The DTD has scope for a third integer if/when more complex sets of mixed channels are standardised.

The author [12] has a personal interest in hyperambisonics (the manipulation of four-, five-dimensional, etc. soundfields) and the experimental code for this should be added, once tied.

5 Acknowledgements

A special thanks to the SoX development team. Not just for SoX, but also for their patient listening to a the various feature requests which were quite exotic compared with normal audio use, and their implementation —without which this work would not have been possible.

Philip Cotterell and Fons Adriaensen kindly checked (and corrected) the early spherical harmonic equations. The former provided those for upto sixteenth order.

Richard Dobson and Richard Furse for some interesting discussions. The former for putting me right about WAVEX files, but most of all for persuading me that adding a GUI was do-able.

Franz Zotter for rotation matrix information for 90° pitches, used in ‘shake’.

Two anonymous reviewers, whose suggestions certainly improved this paper.

6 Acronyms

Acronyms are generally defined in the text. The commoner ones, which are not, are defined here.

DTD Document Type Definition [18]

GUI graphical user interface (a window/icon/mouse/pointer access to a program, rather than a terminal)

GUID globally unique identifier (Microsoft’s implementation of UUIDs).

N3D [14], pp. 155-157, esp. table 3.2 on p. 156

PDF Adobe’s Portable Document Format

UUID universally unique identifier (**man uuidgen** for basic information).

⁵Man is the only animal that can both laugh and cry: for he can see both how things are and how they might have been.

XML Extensible Markup Language [18]

References

- [1] Cotterell, Philip & Michael Chapman. 2009. "Towards a comprehensive account of valid ambisonic transformations". Paper submitted to the First Ambisonics Symposium, IEM. Institute of Electronic Music and Acoustics, Graz, Austria, June 25-27, 2009.
- [2] Perl <http://www.perl.org/>
- [3] SOund eXchange
<http://sox.sourceforge.net/>
- [4] See <http://ambisonics.ch/standards/filetypes/concord>.
- [5] *Apple Core Audio Format Specification 1.0*, Apple Inc. (1 Infinite Loop, Cupertino, CA 95014, USA). 2006-03-08. 62pp.
(Also at <http://developer.apple.com/documentation/MusicAudio/Reference/CAFSpec/CAFSpec.pdf>)
- [6] WavPack and WvUnpack
<http://www.wavpack.com/>.
- [7] Fons Adriaensen, *pers.comm.*
- [8] Open Sound Control,
<http://opensoundcontrol.org>
- [9] See section 4.7 of Fons Adriaensen, October 2008, "AmbDec - 0.4.0 User Manual"
<http://www.kokkinizita.net/linux/audio/downloads/ambdec-manual.pdf>
- [10] Franz Zotter, 2009, "Spherical Harmonics Rotation Matrices. Rotation of Spherical Harmonics in R^3 ".
<http://ambisonics.iem.at/xchange/format/docs/spherical-harmonics-rotation>
- [11] Gerzon, Michael A. & Geoffrey J. Barton. "Ambisonic Decoders for HDTV", Audio Engineering Society Preprint, from the 92nd Convention 1992 March 24-27 Vienna. Available for purchase-download from www.aes.org.
- [12] Chapman, Michael. 2008. "New Dimensions for Ambisonics". Audio Engineering Society Convention Paper 7478. 124th Convention 2008 May 17-20, Amsterdam.
(Also at <http://mchapman.com/amb/hyper.>)
- [13] Dave Malham, 2003, "Higher order Ambisonic systems",
http://www.york.ac.uk/inst/mustech/3d_audio/higher_order_ambisonics.pdf (abstracted from "Music in Space — Space in Music", Mphil thesis, University of York, 2003).
- [14] Daniel, Jérôme. "Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia" Thèse de doctorat de l'Université Paris 6, 31 juillet 2001.
(Also at: <http://gyronymo.free.fr/audio3D/downloads/These.pdf>)
- [15] See page 6, here. For examples, etc., see <http://ambisonics.ch/standards/channels/>.
- [16] <http://www.tcl.tk/>
- [17] <http://search.cpan.org/~gaas/Tkx-1.04/Tkx.pm>
- [18] Harold, Elliotte Rusty & W. Scott Means, 2002. "XML in a Nutshell", second edition. xvii+613 pp. O'Reilly, Sebastopol, USA.

(URLs accessed February 2009.)

Appendices

Download and licence

Ambisuite consists of a GUI called by the command **ambisuite**, and command line utilities:

- ambman** — the basic manipulation (conversion and transformation) program
- ambcalc** — prints out channel values for a given azimuth and elevation
- ambchan** — converts channel names/notations, and likewise for groups of channels
- ambinfo** — reads a WAVEX header and outputs to terminal
- amb2caf** — converts a classic **.amb** file to a **.caf** file with metadata
- wav2amb** — 'corrects' the headers on Wave Format Extensible files creating a **.amb** file

It has a recent Sourceforge page (<http://ambisuite.sourceforge.net/>)

though most of the material still resides at <http://mchapman.com/amb/soft/index>. The Sourceforge site⁶ also contains the ‘forum’ for help, bug reports and feature requests.

It requires Perl (almost any version) for all operations, the audio manipulations require SoX (latest version) and the GUI requires the Tk toolkit and Perl Tkx (see above for references giving URLs).

Ambcalc will present a nice PDF output (see figure 1) as well as the terminal output. (A L^AT_EX file is optionally produced and can be processed with `pdflatex` or similar.).

utility	Perl	SoX
ambman	yes	yes
ambcalc	yes	no
ambchan	yes	no
ambinfo	yes	no
amb2caf	yes	yes
wav2amb	yes	yes

Table 2: Dependencies

It is released under the GNU General Public License, version 3.

Note

This paper was written as some of the code was being finalised. The released version 0.6.0 may show some slight variation. (Screenshots, etc. are from version 0.5..)*

Ambisonics

Unfortunately it is difficult to give an appropriate reference explaining basic ambisonics. Malmham [13] though excellent covers material most likely well known to most ambisonic users. For the newcomer, reading this paper, the differences in channel notation, sequence and weightings is likely to be a hurdle.

The basics used here are explained below.

channels

An ambisonics signal set represents sound from all directions. That is from the full 360° for two dimensions (*pantophony*) or a full sphere (*periphony*). Hybrids of these are possible (see Mixed Order Sets below.) The accuracy of the representation is increased by adding more channels, these are added in discrete groups or *degrees*.

The fact that a full soundfield is represented, means that transformations [1] –such as rotating the soundfield– are possible.

An ambisonic signal set is described as being of a certain *order*.⁷

Zero-order is possible, it comprises one channel and is effectively omni-directional mono.

For practical ambisonic files it is necessary to add first– (second–, etc) degree signals. Thus a third-order file contains zero–, first–, second– and third-degree signals.

For pantophony each additional degree consists of two channels. So the channel count is $2l + 1$, where l is the order.

For periphony the number of channels in each degree is different (3, 5, 7, ...) and the channel count is $(l + 1)^2$.

The channels within a degree are referred to by their order (not to be confused with ambisonic order) m , with $-l \geq m \geq l$. Pantophonic sets only contain channels with $m = \pm l$. It is convenient to give a unique integer to each channel (not least for ordering them in a file), the *ambisonic channel number* (ACN), given by $l(l + 1) + m$.

ACNs 0 to 15 correspond, respectively, to the FuMa letter codes W YZX VTRSU QOMKLN. (There are no letter codes for higher channels. FuMa also applies a (non-normalised) ‘weighting’ (an amplitude factor) to each channel [13]).

A generalised notation for channels uses l and m , with each channel designated B_l^m . If ACNs are used the simpler B_n where n is the ACN can be used.

mixed order sets

Traditionally mixed order signal sets have the lower degree(s) in periphony and the higher one(s) in pantophony. This gives accurate horizontal representation and a more limited representation of ‘height’ information. This corresponds with the soundstage for performed music.

Classically these were periphonic (three-dimensional in lower degrees) with pantophony in higher degrees. Ambisuite (and the DTD it uses) currently recognises this system. However other mixes are possible, for example second-order with all components except channel 6 (R). This eight channel signal set is all that is

⁶https://sourceforge.net/forum/forum.php?forum_id=898927.

⁷Here *order* refers to the entire signal set/file, whilst *degree* refers to each spherical harmonic degree. Previously *order* has been used by some authors for both terms.

needed, for example, for playback on a cuboidal rig⁸ and also has the convenience of being losslessly compressible in a FLAC file. These more complex mixed order sets need three integers to describe them. The current favoured notation is (H, V, P) , though this remains to be standardised and published.

For a more detailed explanation of these signal set terms, and their usage see Cotterell & Chapman [1].

⁸Channel 6 is a “discarded ambiguous harmonic”, see Richard Furse’s *First and Second Order Ambisonic Decoding Equations* (at www.muse.demon.co.uk/ref/speakers.html) (and unlike channel 8 (U) has no potential use if it is desired to yaw the soundfield.)